

# A SPARSITY FOR DECOMPOSING A SYMMETRIC MATRIX

JENN-CHING LUO

**Columbia University, New York, NY 10027, U.S.A.**

*(Received July 1992)*

**Abstract**—This paper studies a sparse configuration for a new class of decomposition derived by the author. Based upon the sparsity, the inverse of a sparse and symmetric matrix and the solution of a system of linear equations can be economically computed. Numerical examples are included.

## 1. INTRODUCTION

Most problems in many disciplines written in differential equations or in integral equations may be discretized by the finite element/difference method. For example, the numerical solution of a differential equation usually begins with a decomposition of the domain into a number of subdomains or elements, then to construct the approximations in a piecewise manner over each subdomain by a set of appropriate trial functions. This trial function used in the discretization procedure can then also be defined in a piecewise manner in the various subdomains. This type of piecewise trial functions is advantageous to make their value zero everywhere except in the subdomains immediately adjacent to this element. This, as is well-known, will give a sparse matrix in the final approximate algebraic equations. The sparse system is an important branch in the field of scientific and engineering computing. Certainly, an economical solution of the sparse system is most desired.

The main difference between the sparse solver and the general solver is that the sparse solver does not operate zero fill-ins. Both iterative and direct methods may be applied to sparse systems. However, applications of the iterative method to sparse systems are easier than the direct method. In the categories of iterative method, applications to sparse systems emphasize data access schemes; while in the scope of direct method, a special sparsity is usually required for a particular method so that the computing streams may avoid the operation of zero fill-ins. For example, the Gaussian elimination is well suited in a banded configuration. Studying a sparsity is necessary for a direct method.

This paper will study a sparse configuration for the author's decomposition [1,2], so that the complexity in inverting a nonsingular sparse matrix and in solving a system of linear equations may be reduced. The author's decomposition is one kind of direct method, and provides a special feature in inverting a nonsingular matrix  $[A]$  directly into  $[A]^{-1}$ . The inverse of  $[A]$  is written as:

$$[A]^{-1} = \begin{cases} [L][L]^T, & \text{if } [A] \text{ is symmetric and positive definite,} \\ [L][D][L]^T, & \text{if } [A] \text{ is symmetric and indefinite,} \\ [L][D][U], & \text{if } [A] \text{ is asymmetric,} \end{cases}$$

in which  $[\overline{L}]$  is a normalized lower triangular matrix,  $[L]$  is a lower triangular matrix with unit diagonal coefficients,  $[D]$  is a diagonal matrix, and  $[U]$  is an upper triangular matrix. This paper is confined to the case where  $[A]$  is symmetric. Generally,  $[A]^{-1}$  is a full matrix, even though  $[A]$  is in a sparse configuration. The author's decomposition writes  $[A]^{-1}$  in the form of  $[A]^{-1} = [L][D][L]^T$ , so that  $[A]$  and  $[L]$  have a possibility of being in the same sparse configuration. The purpose of this paper is to study such a sparsity.

## 2. SPARSE CONFIGURATIONS

As mentioned in the previous section, the author's decomposition may decompose a nonsingular matrix  $[A]$  directly into  $[A]^{-1}$ , and the inverse of  $[A]$  is written in the form of  $[L][D][L]^T$ . The procedure for decomposing  $[A]$  into  $[A]^{-1} = [L][D][L]^T$  is as follows [1]:

For  $j = n \rightarrow 1$  with step  $(-1)$ , do the following

(a) For  $i = (j + 1) \rightarrow n$  with step 1, do

$$L_{ij} \leftarrow L_{ij} + \sum_{k=i+1}^n L_{ki} * L_{kj}; \quad (1)$$

$$(b) D_{jj} \leftarrow \frac{1}{D_{jj} - \sum_{k=j+1}^n L_{kj} * L_{kj} * D_{kk}}; \quad (2)$$

(c) For  $i = n \rightarrow (j + 1)$  with step  $(-1)$ , do

$$L_{ij} \leftarrow -L_{ij} * D_{ii} - \sum_{k=j+1}^{i-1} L_{ik} * L_{kj} * D_{kk}. \quad (3)$$

Since  $[A]$  is confined to be symmetric, considering the lower triangular part and diagonal coefficients is sufficient. After the computations of Eqs. (1)–(3), the lower triangular part of  $[A]$  is  $[L]$ , and the diagonal part of  $[A]$  is  $[D]$ . The solution of  $[A]\{X\} = \{B\}$  is computed by matrix-vector multiplications as

$$\{X\} = [L][D][L]^T \{B\}. \quad (4)$$

It can be realized that this approach provides a more convenient way of applications than the usual methods for the solution of a system of linear equations, because the decomposed results have represented the inverse. This procedure in Eqs. (1)–(3) is computed from  $j = n$  to  $j = 1$ . Let  $\beta_j$  be the half bandwidth of column  $j$  such that  $A_{ij} = 0$  where  $i > j + \beta_j$  (we only consider the lower triangular part of  $[A]$ ). This leads to the following fact.

**FACT 1.** In the  $j^{\text{th}}$  column of a sparse matrix  $[A]$ ,  $A_{ij} = 0$  where  $i > j + \beta_j$ . ■

When dealing with sparse systems, we have to find the sufficient condition such that  $L_{ij} = 0$  where  $i > j + \beta_j$ .

**LEMMA 1.** If  $L_{ik} = 0$  where  $i > j + \beta_j$  and  $j + 1 \leq k \leq j + \beta_j$ , then  $L_{ij} = 0$ .

**PROOF.** The sparsity is determined by Eqs. (1) and (3).

- (a) First, let us discuss the condition required by Eq. (1). Because of sharing with the same computer memories, the  $j^{\text{th}}$  column  $L_{ij}$  in the right side of Eq. (1) are identical to the  $j^{\text{th}}$  column  $A_{ij}$ . By Fact 1,  $L_{ij} = A_{ij} = 0$  when  $i > j + \beta_j$ , which says the first term  $L_{ij}$  in the right side of Eq. (1) is zero when  $i > j + \beta_j$ . Furthermore, the subscript  $k$  in the second term of the right side of Eq. (1) is greater than  $i$ , i.e.,  $k > i > j + \beta_j$ , so that Fact 1 says that  $L_{kj} = A_{kj} = 0$ . This says that the second term in the right side of Eq. (1) is zero when  $i > j + \beta_j$ , and implies that Eq. (1) is zero when  $i > j + \beta_j$ .
- (b) Second, let us consider the sparsity in Eq. (3). The  $j^{\text{th}}$  column  $L_{ij}$  in the right side of Eq. (3) was determined by Eq. (1), so that the first term in the right side of Eq. (3) is zero when  $i > j + \beta_j$ . Since  $L_{kj} = 0$  where  $k > j + \beta_j$ , the upper bound of  $k$  in Eq. (3) may be rewritten as  $\min(i - 1, j + \beta_j) = j + \beta_j$  since  $i > j + \beta_j$ . The second term in the right side of Eq. (3) becomes

$$\sum_{k=j+1}^{j+\beta_j} L_{ik} * L_{kj} * D_{kk}. \quad (5)$$

Since the condition that  $L_{ik} = 0$  where  $i > j + \beta_j$  and  $j + 1 \leq k \leq j + \beta_j$ , the result in Eq. (5) is zero when  $i > j + \beta_j$ . This says, when  $i > j + \beta_j$  and  $j + 1 \leq k \leq j + \beta_j$ , that the second term in the right side of Eq. (3) is zero, and implies that Eq. (3) is zero. This proves the lemma. ■



in which the first column set indicates the connectivity among the triangular non-zero fill-ins. This sparsity has a shape of stairs, and can be applied to a sparse matrix and its inverse. Then, the procedure shown in Eqs. (1)–(3) may be modified according to the stairs-shape sparsity so as to deal with a sparse system.

### 3. COMPUTATIONAL CONSIDERATIONS

Sparse systems have a special feature with a large amount of zero fill-ins. Storing and operating such zero fill-ins may be avoided by an available sparsity. This work has proved that a shape of stairs may provide a sparsity for the author's decomposition, so that the number of arithmetic operations and computer memories may be reduced. For decomposing a sparse matrix  $[A]$  with a shape of stairs into  $[A]^{-1} = [L][D][L]^T$ , Eqs. (1)–(3) may be written as:

For  $j = n \rightarrow 1$  with step  $(-1)$ , do the following

(a) For  $i = (j + 1) \rightarrow (j + \beta_j)$  with step 1, do

$$L_{ij} \leftarrow L_{ij} + \sum_{k=i+1}^{\min(i+\beta_i, j+\beta_j)} L_{ki} * L_{kj}; \quad (10)$$

$$(b) D_{jj} \leftarrow \frac{1}{D_{jj} - \sum_{k=j+1}^{j+\beta_j} L_{kj} * L_{kj} * D_{kk}}; \quad (11)$$

(c) For  $i = (j + \beta_j) \rightarrow (j + 1)$  with step  $(-1)$ , do

$$L_{ij} \leftarrow -L_{ij} * D_{ii} - \sum_{\substack{k=j+1 \\ k+\beta_k \geq i}}^{i-1} L_{ik} * L_{kj} * D_{kk}, \quad (12)$$

in which  $\beta_j$  is the half bandwidth of the  $j^{\text{th}}$  column. Then, with the stairs-shape sparsity, the procedure for computing  $\{X\} = [L][D][L]^T\{B\}$  is written as:

(a) For  $j = 1 \rightarrow n$  with step 1, do

$$X_j \leftarrow X_j + \sum_{i=j+1}^{j+\beta_j} L_{ij} * X_i; \quad (13)$$

(b) For  $j = 1 \rightarrow n$  with step 1, do

$$X_j \leftarrow X_j * D_{jj}; \quad (14)$$

(c) For  $j = n \rightarrow 1$  with step  $(-1)$ , do

$$\begin{aligned} &\text{For } i = (j + 1) \rightarrow (j + \beta_j), \text{ do} \\ &X_i \leftarrow X_i + L_{ij} * X_j, \end{aligned} \quad (15)$$

in which  $\{B\}$  and  $\{X\}$  share the same computer memories. No additional computer memories are required in Eqs. (10)–(15).

The subscript  $k$ , in Eq. (12), ranging from  $(j + 1)$  to  $(i - 1)$  is with the condition  $k + \beta_k \geq i$ , i.e., the term  $L_{ik} * L_{kj} * D_{kk}$  is computed if  $k + \beta_k \geq i$ . The second term in the right side of Eq. (12) consists of not only arithmetic operations but also a logical test. The costs for such logical tests may be saved by introducing a temporary array  $\{S\}$  of order  $n$ . Then, the procedure in Eqs. (10)–(12) may be rewritten as:

For  $j = n \rightarrow 1$  with step  $(-1)$ , do the following

(a) For  $i = (j + 1) \rightarrow (j + \beta_j)$  with step 1, do

$$S_i \leftarrow L_{ij} + \sum_{k=i+1}^{\min(i+\beta_i, j+\beta_j)} L_{ki} * L_{kj}; \quad (16)$$

$$(b) D_{jj} \leftarrow \frac{1}{D_{jj} - \sum_{k=j+1}^{j+\beta_j} S_k * S_k * D_{kk}}; \quad (17)$$

(c) For  $k = (j + \beta_j) \rightarrow (j + 1)$  with step  $(-1)$ , do

$$L_{kj} \leftarrow -S_k * D_{kk}, \quad (18)$$

$$L_{ij} \leftarrow L_{ij} + L_{ik} * L_{kj}, \quad i = k + 1 \rightarrow k + \beta_k, \quad (19)$$

in which Eqs. (18) and (19) are with less operations than Eq. (12).

Another alternative to Eqs. (10)–(15) may be introduced by the concept of blocks, based upon which Eqs. (1)–(3) may be rewritten as

For  $J = N \rightarrow 1$  with step  $(-1)$ , do the following

(a) For  $I = (J + 1) \rightarrow N$  with step 1, do

$$[L_{IJ}] \leftarrow [L_{IJ}] + \sum_{K=I+1}^N [L_{KI}]^T [L_{KJ}]; \quad (20)$$

$$(b) [D_{JJ}] \leftarrow ([D_{JJ}] - \sum_{K=J+1}^N [L_{KJ}]^T [D_{KK}] [L_{KJ}])^{-1}; \quad (21)$$

(c) For  $I = N \rightarrow (J + 1)$  with step  $(-1)$ , do

$$[L_{IJ}] \leftarrow -[D_{II}] [L_{IJ}] - \sum_{K=J+1}^{I-1} [L_{IK}] [D_{KK}] [L_{KJ}], \quad (22)$$

in which  $N$  is the number of partitions. When dealing with a stairs-shape sparsity, a triangular shape of non-zero fill-ins may be viewed as a submatrix, and a sparse matrix  $[A]$  in the form of (9) may be written as

$$[A] = \begin{bmatrix} [A_{11}] & & & & \\ [A_{21}] & [A_{22}] & & & \\ [A_{31}] & & [A_{33}] & & \\ \vdots & & & \ddots & \\ [A_{N1}] & & & & [A_{NN}] \end{bmatrix}.$$

Then, the procedure for decomposing  $[A]$  into  $[A]^{-1}$  becomes

(a) For  $I = 2 \rightarrow N$ , do

$$[A_{II}] \leftarrow [A_{II}]^{-1}; \quad (23)$$

$$(b) [A_{11}] \leftarrow \left[ [A_{11}] - \sum_{K=2}^N [A_{K1}]^T [A_{KK}] [A_{K1}] \right]^{-1}; \quad (24)$$

(c) For  $I = 2 \rightarrow N$ , do

$$[A_{I1}] \leftarrow -[A_{II}] [A_{I1}]. \quad (25)$$

Eq. (23) is implemented by the element-based procedure as shown in (1)–(3). In Eqs. (24) and (25), matrices  $[A_{KK}]$  and  $[A_{II}]$  are in terms of  $[L][D][L]^T$ . Therefore, computing  $[A_{K1}]^T [A_{KK}] [A_{K1}]$  in Eq. (24) consists of five matrices and computing  $[A_{II}] [A_{I1}]$  in Eq. (25) consists of four matrices. We know that a repeat of execution streams may be saved by some additional computer memories. The computing streams in Eq. (25) is a portion of Eq. (24), so that the costs for computing Eqs. (24) and (25) can be reduced by introducing a temporary computer memory  $[W]$ . Then, the procedure shown in Eqs. (23)–(25) may be rewritten as

(a) For  $I = 2 \rightarrow N$ , do

$$[A_{II}] \leftarrow [A_{II}]^{-1}; \quad (26)$$

(b) For  $I = 2 \rightarrow N$ , do

$$[W_I] \leftarrow [A_{I1}], \quad (27)$$

$$[A_{I1}] \leftarrow -[A_{II}] [A_{I1}], \quad (28)$$

$$[A_{11}] \leftarrow [A_{11}] + [W_I]^T [A_{I1}]; \quad (29)$$

$$(c) [A_{11}] \leftarrow [A_{11}]^{-1}. \quad (30)$$

After decomposition, the results obtained from the block-based procedure represent

$$[D] = \begin{bmatrix} [A_{11}] & & & & \\ & [A_{22}] & & & \\ & & [A_{33}] & & \\ & & & \ddots & \\ & & & & [A_{NN}] \end{bmatrix},$$

$$[L] = \begin{bmatrix} [I] & & & & \\ [A_{21}] & [I] & & & \\ [A_{31}] & & [I] & & \\ \vdots & & & \ddots & \\ [A_{N1}] & & & & [A_{NN}] \end{bmatrix},$$

where  $[I]$  is the identity matrix. Then, the solution of  $[A]\{X\} = \{B\}$  is computed by the following procedure:

- (a)  $\{X_1\} \leftarrow \{X_1\} + \sum_{I=2}^N [A_{I1}]^T \{X_I\};$
- (b) For  $J = 1 \rightarrow N$ , do  
 $\{X_J\} \leftarrow [A_{JJ}] \{X_J\};$
- (c) For  $J = 2 \rightarrow N$ , do  
 $\{X_J\} \leftarrow \{X_J\} + [A_{J1}] \{X_1\},$

in which the partition of  $\{X\}$  is consistent with matrix  $[A]$ .

#### 4. EXAMPLES AND DISCUSSIONS

In this section, we will use a simple example to illustrate the element-based procedure and the block-based procedure. Two computer codes in FORTRAN 77 based upon the element-based procedure and the block-based procedure, respectively, are developed for the inverse of a sparse symmetric matrix and the solution of a system of linear equations. The example matrix  $[A]$  of order  $(7 \times 7)$  is written as

$$\begin{bmatrix} 1 & & & & & & \\ & 2 & 3 & & & & \\ & 4 & 5 & 6 & & & \\ & 7 & & & 8 & & \\ & 9 & & & & 2 & \\ -3 & & & & -5 & 6 & \\ 7 & & & & & & 8 \end{bmatrix}. \quad (31)$$

Then, the element-based procedure decomposes  $[A]$  into the following result

$$\begin{bmatrix} 0.178 & & & & & & \\ -1.143 & -0.857 & & & & & \\ 0.286 & -0.833 & 0.167 & & & & \\ -0.875 & & & 0.125 & & & \\ 3.000 & & & & -0.462 & & \\ 3.000 & & & & 0.833 & 0.167 & \\ -0.875 & & & & & & 0.125 \end{bmatrix}, \quad (32)$$

which represents

$$[D] = \begin{bmatrix} 0.178 & & & & & & \\ & -0.857 & & & & & \\ & & 0.167 & & & & \\ & & & 0.125 & & & \\ & & & & -0.462 & & \\ & & & & & 0.167 & \\ & & & & & & 0.125 \end{bmatrix}, \quad (33)$$

$$[L] = \begin{bmatrix} 1.000 & & & & & & \\ -1.143 & 1.000 & & & & & \\ 0.286 & -0.833 & 1.000 & & & & \\ -0.875 & & & 1.000 & & & \\ 3.000 & & & & 1.000 & & \\ 3.000 & & & & 0.833 & 1.000 & \\ -0.875 & & & & & & 1.000 \end{bmatrix}. \quad (34)$$

It can be verified that  $[L][D][L]^T$  is the inverse of  $[A]$ . Similarly, when dealing with the block-based procedure, the matrix  $[A]$  is partitioned into the form

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 4 \\ 7 \\ 9 \\ -3 \\ 7 \end{bmatrix} & \begin{bmatrix} 3 \\ 5 \\ 6 \end{bmatrix} & \text{sym.} \\ & [8] & \\ & \begin{bmatrix} 2 \\ -5 \\ 6 \end{bmatrix} & \\ & & [8] \end{bmatrix}, \quad (35)$$

which can be decomposed into

$$\begin{bmatrix} [0.178] & & & & \text{sym.} \\ \begin{bmatrix} -1.143 \\ 0.286 \end{bmatrix} & \begin{bmatrix} -0.857 \\ -0.833 \quad 0.167 \end{bmatrix} & & & \\ \begin{bmatrix} -0.875 \end{bmatrix} & & [0.125] & & \\ \begin{bmatrix} 3.000 \\ 3.000 \end{bmatrix} & & & \begin{bmatrix} -0.462 \\ 0.833 \quad 0.167 \end{bmatrix} & \\ \begin{bmatrix} -0.875 \end{bmatrix} & & & & [0.125] \end{bmatrix}. \quad (36)$$

It is no surprise that the results obtained from the element-based procedure and the block-based procedure as shown in Eqs. (32) and (36) are identical. The result in Eq. (36) represents

$$[D] = \begin{bmatrix} [0.178] & & & & \\ & \begin{bmatrix} -0.857 \\ -0.833 \quad 0.167 \end{bmatrix} & & & \\ & & [0.125] & & \\ & & & \begin{bmatrix} -0.462 \\ 0.833 \quad 0.167 \end{bmatrix} & \\ & & & & [0.125] \end{bmatrix},$$

$$[L] = \begin{bmatrix} [1.000] & & & & \\ \begin{bmatrix} -1.143 \\ 0.286 \end{bmatrix} & \begin{bmatrix} 1.000 \\ 0.000 \quad 1.000 \end{bmatrix} & & & \\ \begin{bmatrix} -0.875 \end{bmatrix} & & [1.000] & & \\ \begin{bmatrix} 3.000 \\ 3.000 \end{bmatrix} & & & \begin{bmatrix} 1.000 \\ 0.000 \quad 1.000 \end{bmatrix} & \\ \begin{bmatrix} -0.875 \end{bmatrix} & & & & [1.000] \end{bmatrix}.$$

It also can be verified in the block-based procedure that  $[L][D][L]^T$  is the inverse of  $[A]$ . Using the stairs-shape sparsity, a nonsingular sparse matrix  $[A]$  can be easily and efficiently inverted into  $[A]^{-1} = [L][D][L]^T$  in which matrix  $[L]$  is also in the same stairs-shape sparsity.

The author's decomposition provides a special feature in inverting a nonsingular matrix, which can decompose a nonsingular matrix  $[A]$  directly into  $[A]^{-1}$ . The inverse of a nonsingular matrix is very useful in many scientific and engineering computations. Especially, as cited in [3], explicitly inverting the left side matrix of  $[A]\{X\} = \{B\}$  may take advantage of multiprocessors when solving  $\{X\}$ . This point can be easily realized, because if  $[A]^{-1}$  is available, then the operations for  $\{X\} = [A]^{-1}\{B\}$  can be efficiently distributed onto employed processors. The author's decomposition can provide an efficient way of inverting a nonsingular matrix. Some outstanding performances of the author's decomposition on an Alliant/FX8 computer for the solution of systems of linear equations were reported in [1,2]. To sparse systems, high efficient parallel performances can also be expected, for example, the computing streams in Eqs. (23) and (25) can be fully parallelized and the matrix operations in Eq. (24) can also be distributed onto employed processors.

Some characteristics of this approach may be summarized as: (1) minimal computer memories: it can be seen, for example from Eqs. (10)–(15), that no additional computer memories are required; (2) the same sparse configuration for  $[A]$  and  $[A]^{-1}$ : this is a special advantage of the author's decomposition.  $[A]^{-1}$  may be a full matrix, even though  $[A]$  is sparse. However, by the author's decomposition,  $[A]^{-1}$  is written in the form of  $[A]^{-1} = [L][D][L]^T$  such that  $[L]$  may be in the same stairs-shape sparsity; (3) a high potential for parallel computations: it can be seen, for example from Eqs. (23)–(25), that the computing streams may be distributed onto employed processors. The stairs-shape sparsity provides an important concept for the inverse of a nonsingular matrix, and can be a fundamental tool in scientific and engineering computing.

#### REFERENCES

1. J.-C. Luo, A new class of decomposition for symmetric systems, *Mechanics Research Communications* **19**, 159–166 (1992).
2. J.-C. Luo, A new class of decomposition for inverting asymmetric and indefinite matrices, *Computers & Mathematics with Applications* (to appear).
3. I.S. Duff, A.M. Erisman and J.K. Reid, *Direct Methods for Sparse Matrices*, Oxford, (1986).